UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/584,167	06/22/2006	Takaharu Hamada	OMOR-0012	5067

23377 7590 05/11/2010

WOODCOCK WASHBURN LLP
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
HAYIM, SAMUEL E	
ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
05/11/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/584,167	HAMADA, TAKAHARU	
	Examiner	Art Unit	
	SAMUEL HAYIM	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 June 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *06/22/2006 02/09/2009*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. The instant application having Application No. 10584167 filed on June 22nd, 2006 where claims 1-12 are presented for examination by the examiner.

Examiner Notes

2. Examiner cites particular columns and line numbers in the references as applied to the claims below for the convenience of the applicant. Although the specified citations are representative of the teachings in the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested that, in preparing responses, the applicant fully consider the references in entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the examiner.

Oath/Declaration

3. The applicant's oath/declaration has been reviewed by the examiner and is found to conform to the requirements prescribed in **37 C.F.R. 1.63.**

Priority

As required bye **M.P.E.P. 201.14(c)**, acknowledgement is made of applicant's claim for priority based on applications filed on December 25, 2003 (Japan 2004-429897).

Receipt is acknowledged of papers submitted under 35 U.S.C. 119(a)-(d), which papers have been placed of record in the file.

However, to overcome a prior art rejection, applicant(s) must submit a translation of the foreign priority papers in order to perfect the claimed foreign priority because said papers has not been made of record in accordance with 37 CFR 1.55. See MPEP § 201.15.

Drawings

4. The applicant's drawings submitted are acceptable for examination purposes.

Information Disclosure Statement

5. As required by M.P.E.P. 609, the applicant's submissions of the Information Disclosure Statement dated 2/09/2009 and 6/22/2006 are acknowledged by the examiner and the cited references have been considered in the examination of the claims now pending; However, several references cited in both IDS' fails to comply with 37 CFR 1.98(a)(3) because it does not include a concise explanation of the relevance, as it is presently understood by the individual designated in 37 CFR 1.56(c) most knowledgeable about the content of the information, of each patent listed that is not in the English language. Therefore, references the following references are not considered:

a. FUJITSU LTD., Manual for OVIS/S LATOX-F (FACOM safety test system package, instructions: OVIS/S LATOX-F), Jan. 31, 1987, pages 1-2, first issue, Japan.

b. SEI (Kiyoshi) MURAKAMI et al, Tokushu Iyakuhin Seizogyo ni Okeru keisoku Seigyo Joho System:Iyakuhin Keisoku Seigyo System to Senjo Variation, Hitachi Hyouron, 1996,04,01, Vol. 78 No. 4. Japan

The references have been placed in the application file, but the information referred to therein has not been considered.

Abstract

6. The abstract of the disclosure is objected to because it is more than one paragraph, Correction is required. See MPEP § 608.01(b).

Specification

7. The applicant's specification is acceptable for examination purposes.

Double Patenting

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., In re Berg, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); In re Goodman, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); In re Longi, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); In re Van Ornum, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); In re Vogel, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and In re Thorington, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Claims 1-2 are provisionally rejected for double patenting over patent number 7,349,810

Claims 1 and 2 is compared to claims 1 and 5 of patent number 7,349,810 in the following table:

Application 11/408,843	Patention 7,349,810
	Claim 1 - A safety test support system for managing data concerning at least one of medicines, pharmaceuticals, agricultural chemicals, food additives or other chemical safety tests on a living body to which at least one of a medicine, pharmaceutical, agricultural chemical, food additive or other chemical has been given, the data including the condition of the living body, measuring of at least one of the living body's weight, food consumption, water consumption, urinary output, biochemical test results, hematological test results, clinical symptom observation results pathological opinions based upon findings and/or observations about the living body, the system comprising:
Claim 1- A computer software program article comprising a **storage medium having stored thereon a computer software program which when executed causes a detection of whether or not there have been one or more specific changes in one or more application programs running on a computer system, said computer software program comprising:**	**first program storage for storing application programs for the safety tests, the programs being functionally partitioned according to data items and/or operations;** an activating means for activating at least one program selected from the application programs; second program storage for storing check programs each for detecting a change in one of the application programs during the system operation;
inspection scenarios, associated with each of said application programs and stored on said storage medium, for detecting whether or not there have been specific changes in each of said application programs;	**and an inspection conducting means for detecting changes in the application programs by sequentially executing the stored check programs in response to an inspection conducting signal wherein the**

	safety test support system is constructed so that:
and an inspection scenario program, for detecting whether or not there has been a specific change in an application program, by running said associated specific application program according to said associated inspection scenario, and for outputting detection results in association with said user name and said application program.	**a) if a change that does not affect the system operation is detected in one of the application programs during the system operation, the associated check program ignores the change;**
	and b) if a change that affects the system operation is detected in one of the application programs during the system operation, the associated check program regards the change as a change
	Claim 5- The safety test support system according to claim 1 and further comprising:
	a displaying means for displaying a list of the application programs stored in the first program storage;
Claim 2 - **A computer software program article as set forth in Claim 1, wherein: said one or more application programs includes an application program executor for displaying the other application programs to the user selectively and executably**	**a program selecting means for selecting one or more of the displayed application programs;**
	the activating means being adapted to activate the selected application program or programs;
	an activation confirming means for confirming at predetermined intervals whether the activated application program or any of the activated application programs is active;
Claim 1 (cont'd) - **an authentication program for authenticating a user prior to**	**and a user authenticating means for requesting at predetermined intervals that**

the execution of at least one of said application programs, and for associating the user in the aforementioned authentication to an application program that will be run later;	**authentication information on the user of the support system be input while the activation confirming means is confirming whether the activated program or any of the activated programs is active**

The program storage areas (first and second) as disclosed by the 7,349,810 patent correspond to the readable storage medium in the pending application which contains storage areas for both a first program (an application program in the copending application) and a second program (inspection scenario program in the copending application). The program selection means and program activation means as stated in the patented case corresponds to the "running [of] said specific application program." The inspection scenarios of the pending application correspond to the "inspection conducting means" which carry out a verification that the application has not been changed. Furthermore, the user authentication as defined by the pending application is analogous to the user authenticating means as claimed in the patent.

As per claim 2 of the instant application, claim 5 of the patent satisfies each of the limitations found in claim 1 as it contains a display for displaying multiple programs for execution as well as a means for execution.

The combined limitations for claims 1 and 5 of patent 7,349,810 match the limitations of claims 1 and 2 of the instant application.

Claims 4 is compared to claims 2 of patent number 7,349,810 in the following table:

Claim 4 - computer software program article as set forth in Claim 1, wherein:	Claim 2 - the safety test support system according to claim 1 wherein the inspection conducting means inspects the application programs by:
	identifying the application program associated with each of the check programs;
said inspection scenario program inputs a dummy signal into *said application program* in accordance with said inspection scenario to	**inputting a pseudo-signal directly to the identified application program;**
detect a response signal to said inputted dummy signal,	**detecting a response signal responsive to the input pseudo-signal;**
to thereby detect whether or not there has been a specific change in the application program.	**and comparing the detected response signal with a response signal detected before the inspection**

As per claim 3, the claim inherits limitations from claim 1 such as a user verification program that the combined limitations disclosed in patent 7,349,810 does not expressly disclose. However it would obvious to add such a limitation to the existing application verification system.

Bryant-Rich (US 2005/0114643) (hereinafter Rich) discloses a verification program (For example, paragraph 3, the use of the computer is restricted to a specified list of users, the applications are installed on the Personal Computer and the configuration and data used by each application are stored separately for each user. A personal computer often also has a generally usable but very restricted access method such as a "guest" account. A program is configured and

stored separately from other users thereby associating that particular program with that particular user).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the program verification system as described by patent 7,349,810 and add a user verification system as taught by Rich because it would provide for the efficient means for controlling the modification of application programs (see Roth paragraphs 3-4).

The program that is identified as being associated with the check programs (inspection scenarios) is "said program" which is a specific chosen program among a plurality of application programs) (the relationship shown above by corresponding italics).

Claims 8 and 9 are compared to claim 6 of patent 7,349,810 in the following table

Claim 8 - A computer software program article as set forth in Claim 1, wherein said authentication program:	Claim 6 - The safety test support system according to claim 5 and further comprising an
comprises an authentication update requesting article for requesting the input of user authentication data	**authentication information holding means for holding validity information on the user authentication by the user authenticating means, wherein,**
at each specific time interval;	**when each of the selected application programs is activated,**
and if the user cannot be authenticated by said authentication update requesting article terminates said application program that is running, associated with the applicable user.	**the execution of the activated application program is started if the user authentication is valid, and the execution of the activated application program is not started if the user authentication is invalid, on the basis of the held validity information**
Claim 9 - A computer software program article as set forth in Claim 1, wherein: said authentication program performs repeat user authentication in response to a user request after an initial user authentication, and associates, with the user involved in said authentication, the application programs currently running, and application programs run thereafter.	

As per claim 8, in order for the authentication information holding means to hold validity information on the user the user must, at some point, input such information.

As per claim 9 the patent 7,349,810 does not expressly disclose said authentication program performs repeat user authentication in response to a user request after an initial user authentication, and associates, with the user involved in said authentication, the application programs currently running, and application programs run thereafter. However it would be obvious to add such limitations by combining Roth (US 5,857,205) in view of Ribot (US 2003/0187993).

Ribot discloses said authentication program performs repeat user authentication in response to a user request after an initial user authentication, (For example, paragraph 12-13, the distributed proxy communications object component is adapted to enable comparison of the contents of the request and the definition of the rights and privileges of the user. The request relates to modification of a management object maintained at a network resource, the organization having a global right to access the network resource. A second client proxy communications object component is preferably adapted to enable forwarding of the request to the server in response to the comparison. Preferably, the request is forwarded only when comparison step determines that the request and the rights and privileges are consistent, Thereby controlling individual authorization of the user each time network resource is accessed).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the program verification system as described by patent 7,349,810 and add a user verification system as taught by Ribot because it would provide for the efficient means for controlling security and access controls during the authentication and authorization of the client organization (see Ribot paragraph 35).

Roth disclose and associates, with the user involved in said authentication, the application programs currently running, and application programs run thereafter (For example, column 1 line 66 to column 2 line 7, in any case, when batch systems encounter undetected errors in the data, the process may or may not respond to the error. In the case where the process is affected by the error, it will either notify the user of a problem in a controlled fashion (if the possibility of that type of error was foreseen) or the process will be forced to a halt (when the error is of an unforeseen nature). The error in the data may also go undetected allowing the process to continue to completion, so that the incorrect data will not be immediately obvious. The errors (changes) output reflect the user associated with the program (under examination).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the program verification system as described by patent 7,349,810 and add a user verification system as taught by Ribot because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

This is a provisional double patenting rejection since the conflicting claims have not yet been patented.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1-3, 6-7 and 10-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bryant-Rich (US 2005/0114643) (hereinafter Rich) in view of Roth (US 5,857,205).

As per claim 1, Rich discloses a computer software program article comprising a storage medium having stored thereon a computer software program which when executed causes a detection of whether or not there have been one or more specific changes in one or more application programs running on a computer system, said computer software program comprising: (For example, figure 1-2, launcher, 14, monitors changes made in applications, 14, running on the computer, 30).

an authentication program for authenticating a user prior to the execution of at least one of said application programs, and for associating the user in the aforementioned authentication to an application program that will be run later; (For example, paragraph 3, the use of the computer is restricted to a specified list of users, the applications are installed on the Personal Computer and the configuration and data used by each application are stored separately for each user. A personal computer often also has a generally usable but very restricted access method such as a "guest" account. A program is configured and stored separately from other users thereby associating that particular program with that particular user).

inspection scenarios, associated with each of said application programs and stored on said storage medium, for detecting whether or not there have been specific changes in each of said application programs; (For example, figure 2, box 58 and 60 are examples of two inspection scenarios that monitor state changes in the application program being executed or the memory device being removed from the computer).

and an inspection scenario program for detecting whether or not there has been a specific change in an application program, by running said associated specific application program according to said associated inspection scenario, (For example, figure 1-2, the launcher, 14 is the inspection scenario program as it detects the state and memory changes with the application that it is executing).

Rich does not expressly disclose for outputting detection results in association with said user name and said application program

However, Roth discloses for outputting detection results in association with said user name and said application program (For example, column 1 line 66 to column 2 line 7, in any case, when batch systems encounter undetected errors in the data, the process may or may not respond to the error. In the case where the process is affected by the error, it will either notify the user of a problem in a controlled fashion (if the possibility of that type of error was foreseen) or the process will be forced to a halt (when the error is of an unforeseen nature). The error in the data may also go undetected allowing the process to continue to completion, so that the incorrect data will not be immediately obvious. The errors (changes) output reflect the user associated with the program (under examination).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

As per claim 2, Rich discloses computer software program article as set forth in claim 1, wherein: said one or more application programs includes an application program executor for displaying the other application programs to the user selectively and executably (For example, figure 3, the launcher, 14, displays a list of applications, 16-16'', to the user for selection).

As per claim 3, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein: said one or more application programs is a plurality of application programs requiring validations in order to fulfill specific standards under identical policies; said inspection scenarios are for detecting whether or not changes in each application program are to the degree that allows execution of the application program without performing a re-validation; and said inspection scenario program instructs said computer system to display the detection result, when, in accordance with said inspection scenario, a change of an extent greater than that wherein execution is allowed without performing validations in detected.

However, Roth discloses wherein: said one or more application programs is a plurality of application programs requiring validations in order to fulfill specific standards under identical

policies; (For example, figure 1, files (from multiple programs) are validated to identical standards using file analysis, 14 during process monitoring (execution), 16).

said inspection scenarios are for detecting whether or not changes in each application program are to the degree that allows execution of the application program without performing a re-validation; (For example, figure 1, data verification and validation is based on errors (changes) in the files accessed by the applications. Serious anomalies and significant variations effect the ability for the application to run; For example, column 1 line66 to column 2 line 7, In any case, when batch systems encounter undetected errors in the data, the process may or may not respond to the error. In the case where the process is affected by the error, it will either notify the user of a problem in a controlled fashion (if the possibility of that type of error was foreseen) or the process will be forced to a halt (when the error is of an unforeseen nature). The error in the data may also go undetected allowing the process to continue to completion, so that the incorrect data will not be immediately obvious).

and said inspection scenario program instructs said computer system to display the detection result, when, in accordance with said inspection scenario, a change of an extent greater than that wherein execution is allowed without performing validations in detected (For example, figure 1 and column 34 line 29-36, employs a generic approach which is driven by record descriptions (a.k.a. record layouts) which may be created for use in programs which read from or write to these files. This software may be used to profile the contents of files, monitor changes, detect likely areas of erroneous data, generate data domain meta-data, and verify "migrated" information in parallel implementations and similar uses. Figure 1 depicts a series of generated

reports that report on errors (changes) whose threshold is set to any. Therefore, the threshold is immediately passed at the occurrence of the first error, generating the reports).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

As per claim 6, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein: said inspection scenario program is launched as specific time intervals

However, Roth discloses wherein: said inspection scenario program is launched as specific time intervals (For example, column 4 line 42-49, the first part compares record layouts over time to determine if they have changed in ways that would affect the contents of files. The second part performs a generic data item evaluation that obtains a description of the contents of every data item that is identified in the record layouts (a.k.a. data item characteristics), and compares these characteristics over time (where historical information is available). The comparison of file layouts over time occurs over a specific time intervals as random does not exist within computer architectures. Therefore, a specific time interval is set).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

As per claim 7, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein: said inspection scenario program comprises a detection results display step for displaying said detection results on a computer display, and a user input/output step for receiving user input regarding said detection results and for outputting in association with said detection results.

However, Roth discloses wherein: said inspection scenario program comprises a detection results display step for displaying said detection results on a computer display, and a user input/output step for receiving user input regarding said detection results and for outputting in association with said detection results (For example, figure 3C, the detection results are displayed to the user. The user is able to manipulate the files; For example, column 6 line 44-65, The first group of items which the online entry procedure prompts the user for are the "File Identification" items 24 such as the "File Name (Descriptive Name)" and "DSN (Use "0" for GDGs)." The "File Name (Descriptive Name)" is the name of the file in plain language. It serves as an essential piece of system documentation. The "DSN (Use "0" for GDGs)" is the "data set

name" and is the "formal" name used by the method to "catalogue" the file. The "File ID" that is subsequently assigned to each file reference is based primarily on the DSN. The reason for substituting a numerical key in place of the DSN is mainly as a space and time saving measure. A DSN (on the MVS system) can be 44 bytes long, the binary packed numerical file ID occupies only 2 bytes. Another reason for using a File ID alias involves the situation where a file's DSN has to be changed. In such a situation, the File ID can be reassigned independently of the DSN, thus maintaining the continuity of references across file generations. If the file is a generation data group (GDG) the user will follow the DSN with "(0)" to indicate the current version. Entering the DSN of an already specified file entry will cause that entry to be retrieved for maintenance purposes. Files are therefore selected based on the users input).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

As per claim 10, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein said computer software program is for insuring that the results of safety testing have not been falsified or tampered with in said application program; and wherein said application program receives measurement values, which have not been falsified or

tampered with, from a measurement device for safety testing, processes the measurement values, and outputs the results of specific processing.

However, Roth discloses wherein said computer software program is for insuring that the results of safety testing have not been falsified or tampered with in said application program; (For example, column 4 line 10-27, file records are check against baseline versions of the same file thereby insuring that the files have not been tampered or falsified regardless of what the files are for).

and wherein said application program receives measurement values, which have not been falsified or tampered with, from a measurement device for safety testing, processes the measurement values, and outputs the results of specific processing (For example, figure 1 column 3 line 42-49, the data is aggregated for a repository of files to be checked (layouts comparison confirms tamper proofed files). The system is a safety testing system as file integrity is safety checking. The files are processed at, 14; For example, figure 3C, the results are output to the user).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

As per claim 11, Rich discloses an application program inspecting system comprising: an application storage unit for storing one or more mutually-related application programs; (For example, figure 1, application programs are stored in memory 12).

an authentication unit for performing user authentication prior to running at least the first of said application programs, and for associating application programs run thereafter, with the user involved in said authentication; (For example, paragraph 3, the use of the computer is restricted to a specified list of users, the applications are installed on the Personal Computer and the configuration and data used by each application are stored separately for each user. A personal computer often also has a generally usable but very restricted access method such as a "guest" account. A program is configured and stored separately from other users thereby associating that particular program with that particular user).

an inspection scenario storage unit for storing inspection scenarios, associated with each of said application programs, for detecting whether or not there have been specific changes in each of said application programs; (For example, figure 1-2, the system comprises a series of inspection scenarios, boxes 58-60 and 64, which detect specific changes to the application program. The inspection scenarios are stored in memory (figure 1)).

and an inspecting unit for detecting whether or not there have been specific changes in said application programs, through an inspection scenario program executing specific related application programs in accordance with said associated inspection scenarios, (For example, figure 1-2, the launcher, 14 (inspection scenario program) executes the specific related application program and determines changes made to the application programs, 16, using the associated scenarios).

Rich does not expressly disclose and for outputting said detection results in association with said user name and application program.

However, Roth discloses and for outputting said detection results in association with said user name and application program (For example, figure 3C, the output file contains the name of the application program (top left written as 'XXXXXXX') and information of the user as seen in lines 11-20 which contain information regarding the name, birthday, hired time, sex, job type, job title, and job class).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

As per claim 12, Rich discloses a method implemented using a computer for detecting whether or not there have been one or more specific changes in one or more mutually-related application programs on a computer system, (For example, figure 1-2, applications are executed on host computer, 30, where changes are detected relating to specific programs).

comprising: an authentication process for performing user authentication prior to running at least the first of said application programs and for associating, with said application programs run thereafter, the user involved in said authentication; (For example, paragraph 3, the use of the

computer is restricted to a specified list of users, the applications are installed on the Personal Computer and the configuration and data used by each application are stored separately for each user. A personal computer often also has a generally usable but very restricted access method such as a "guest" account. A program is configured and stored separately from other users thereby associating that particular program with that particular user).

and an inspection process for detecting whether or not there has been a specific change in said application program, through the use of an inspection scenario, associated with each application program, for detecting whether or not there has been a specific change in each of the application programs, by an inspection scenario program executing a specific related application program in accordance with said associated inspection scenario, (For example, figure 1-2, the system comprises a series of inspection scenarios, boxes 58-60 and 64, which detect specific changes to the application program. The inspection scenarios are stored in memory (figure 1); For example, figure 1-2, the launcher, 14 (inspection scenario program) executes the specific related application program, 16, and determines changes made to the application programs using the associated scenarios).

Rich does not expressly disclose and outputting the detection results in association with said user name and application program.

However, Roth discloses and outputting the detection results in association with said user name and application program (For example, figure 3C, the output file contains the name of the application program (top left written as 'XXXXXXX') and information of the user as seen in lines 11-20 which contain information regarding the name, birthday, hired time, sex, job type, job title, and job class).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

10. Claims 4-5 and 8-9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bryant-Rich (US 2005/0114643) (hereinafter Rich) in view of Roth (US 5,857,205) in further view of Ribot (US 2003/0187993).

As per claim 4, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein: said inspection scenario program inputs a dummy signal into said application program in accordance with said inspection scenario to detect a response signal to said inputted dummy signal, to thereby detect whether or not there has been a specific change in the application program.

However, Ribot discloses wherein: said inspection scenario program inputs a dummy signal into said application program in accordance with said inspection scenario to detect a response signal to said inputted dummy signal, to thereby detect whether or not there has been a specific change in the application program (For example, paragraph 41-49, the distributed object

CommMgr in, accordance with the present invention inherits from a second object whose only function is to limit the range of operations of the distributed object CommMgr. This second object does not define any operation (or method), but contains the reference of an organization and/or of a BN 4, which specifies the domain of application of the operations, i.e. it is a data object. This second object is resident in server 11 or could be provided on another node of the OMN 16, e.g. in database server 19. Each proxy (CommMgr) distributed by the MD 10 defines exactly the domain of application of offered services. These data cannot be modified by the client organization as the client's terminals have read-only access to the attributes defining the client's profile. So, at the time the client organization requests a first connection, the receiving server of the OMN 16, after authentication, will generate the proxy to be sent to that client organization in function of the services and domains to which the organization has rights. This proxy is now available at the client organization's external terminal. The next request from the client's terminal will access the proxy which enables a comparison of the request and the authorization credentials in the proxy. Depending upon the comparison, the proxy can enable forwarding of the request. For example, if the request and the privileges do not match, the request is aborted and/or any other suitable action is taken to prevent the request being made. On the other hand if the authorization credentials are consistent with the request, the proxy enables forwarding of the request to the server 16. Afterwards, it will not be necessary to verify the rights of the client organization as the fact that the client organization is in possession of a proxy which has allowed the request to proceed to the server will be sufficient for proving the rights of the client organization. The two proxies are added to the interchange between client and server to

protect the integrity of the network resource. The proxies thereby refer to a dummy signal that is used to detect changes (requests for changes to network resources)).

Rich and Ribot are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and control access between the user and the software as taught by Ribot because it would provide for the efficient means for controlling security and access controls during the authentication and authorization of the client organization (see Ribot paragraph 35).

Rich does not expressly disclose and associates, with the user involved in said authentication, the application programs currently running, and application programs run thereafter.

As per claim 5, Rich discloses a computer software program article as set forth in claim 4, wherein: said inspection scenario includes at least data for specifying an application program to be subjected to inspection, (For example, figure 2, the launcher is executed, 54, (control program to detect changes made to application program) then the application specified is executed at box 56).

and data regarding an allowable range regarding the response to said data (For example, figure 2, the responses from the application as it is executed is detected by launcher (boxes 58-60 and 64) that determine the allowable range of changes made to the application).

Rich does not expressly disclose data for inputting as dummy signals into said application program.

However Ribot discloses data for inputting as dummy signals into said application program, (For example, paragraph 41-49, the distributed object CommMgr in, accordance with the present invention inherits from a second object whose only function is to limit the range of operations of the distributed object CommMgr. This second object does not define any operation (or method), but contains the reference of an organization and/or of a BN 4, which specifies the domain of application of the operations, i.e. it is a data object. This second object is resident in server 11 or could be provided on another node of the OMN 16, e.g. in database server 19. Each proxy (CommMgr) distributed by the MD 10 defines exactly the domain of application of offered services. These data cannot be modified by the client organization as the client's terminals have read-only access to the attributes defining the client's profile. So, at the time the client organization requests a first connection, the receiving server of the OMN 16, after authentication, will generate the proxy to be sent to that client organization in function of the services and domains to which the organization has rights. This proxy is now available at the client organization's external terminal. The next request from the client's terminal will access the proxy which enables a comparison of the request and the authorization credentials in the proxy. Depending upon the comparison, the proxy can enable forwarding of the request. For example, if the request and the privileges do not match, the request is aborted and/or any other suitable action is taken to prevent the request being made. On the other hand if the authorization credentials are consistent with the request, the proxy enables forwarding of the request to the server 16. Afterwards, it will not be necessary to verify the rights of the client organization as the fact that the client organization is in possession of a proxy which has allowed the request to proceed to the server will be sufficient for proving the rights of the client organization. The two

proxies are added to the interchange between client and server to protect the integrity of the network resource. The proxies thereby refer to a dummy signal that is used to detect changes (requests for changes to network resources) and data exists within the system to build the proxies between the client and server).

Rich and Ribot are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and control access between the user and the software as taught by Ribot because it would provide for the efficient means for controlling security and access controls during the authentication and authorization of the client organization (see Ribot paragraph 35).

As per claim 8, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein said authentication program: comprises an authentication update requesting article for requesting the input of user authentication data at each specific time interval; and if the user cannot be authenticated by said authentication update requesting article, terminates said application program that is running, associated with the applicable user.

However, Ribot discloses wherein said authentication program: comprises an authentication update requesting article for requesting the input of user authentication data at each specific time interval; (For example, paragraph 49, the user authentication occurs each time a request is made to the network resource. Each request is a specific time interval).

and if the user cannot be authenticated by said authentication update requesting article, terminates said application program that is running, associated with the applicable user (For

example, paragraph 50, the access control decision function (ACDF) (which is the procedure (or set of procedures) that applies the access control rules to each access request so as to determine whether the requested access to a management object should be granted or denied) is implemented by the proxy in accordance with the present invention. The access control enforcement function (ACEF) (which is the procedure (or set of procedures) for enforcing the decisions made by the ACDF) is also performed by the proxy. Authorization is denied when a network resource (management object), or application, is accessed by an unauthorized user thereby halting the application).

Rich and Ribot are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and control access between the user and the software as taught by Ribot because it would provide for the efficient means for controlling security and access controls during the authentication and authorization of the client organization (see Ribot paragraph 35).

As per claim 9, Rich does not expressly disclose a computer software program article as set forth in claim 1, wherein: said authentication program performs repeat user authentication in response to a user request after an initial user authentication.

However, Ribot discloses a computer software program article as set forth in claim 1, wherein: said authentication program performs repeat user authentication in response to a user request after an initial user authentication, (For example, paragraph 12-13, the distributed proxy communications object component is adapted to enable comparison of the contents of the request

and the definition of the rights and privileges of the user. The request relates to modification of a management object maintained at a network resource, the organization having a global right to access the network resource. A second client proxy communications object component is preferably adapted to enable forwarding of the request to the server in response to the comparison. Preferably, the request is forwarded only when comparison step determines that the request and the rights and privileges are consistent, Thereby controlling individual authorization of the user each time network resource is accessed).

Rich and Ribot are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and control access between the user and the software as taught by Ribot because it would provide for the efficient means for controlling security and access controls during the authentication and authorization of the client organization (see Ribot paragraph 35).

Rich does not expressly disclose and associates, with the user involved in said authentication, the application programs currently running, and application programs run thereafter.

However, Roth disclose and associates, with the user involved in said authentication, the application programs currently running, and application programs run thereafter (For example, column 1 line 66 to column 2 line 7, in any case, when batch systems encounter undetected errors in the data, the process may or may not respond to the error. In the case where the process is affected by the error, it will either notify the user of a problem in a controlled fashion (if the

possibility of that type of error was foreseen) or the process will be forced to a halt (when the error is of an unforeseen nature). The error in the data may also go undetected allowing the process to continue to completion, so that the incorrect data will not be immediately obvious. The errors (changes) output reflect the user associated with the program (under examination).

Rich and Roth are analogous art because they are from the same field of endeavor of software execution management.

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the launcher application that controls execution of subsequent applications as described by Rich and output the results of the changes made to the user associated with the program executed as taught by Roth because it would provide for the efficient means for detecting errors and providing data verification throughout the entire computer system (see Roth column 3, line 1-4).

Conclusion

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Samuel Hayim whose telephone number is (571) 270-3370. The examiner can normally be reached on Monday to Friday 8:30 AM to 5:00 PM.

If attempts to reach the above noted Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached at the following telephone number: (571) 272-3695.

The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/SAMUEL HAYIM/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192